

BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

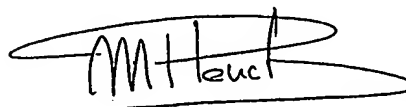
REC'D 04 FEB 2004

WIPO PCT

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 25 NOV. 2003

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets



Martine PLANCHE

DOCUMENT DE PRIORITÉ

PRÉSENTÉ OU TRANSMIS
CONFORMÉMENT À LA
RÈGLE 17.1.a) OU b)

BEST AVAILABLE COPY

INSTITUT
NATIONAL DE
LA PROPRIÉTÉ
INDUSTRIELLE

SIEGE
26 bis, rue de Saint Petersburg
75800 PARIS cedex 08
Téléphone : 33 (0)1 53 04 53 04
Télécopie : 33 (0)1 53 04 45 23
www.inpi.fr

REQUÊTE EN DÉLIVRANCE

page 1/2





Cet imprimé est à remplir lisiblement à l'encre noire

DB 540 0 W / 210502

REMISE DES PIÈCES DATE 15 NOV 2002 LIEU 54 INPI NANCY N° D'ENREGISTREMENT 0214281 NATIONAL ATTRIBUÉ PAR L'INPI DATE DE DÉPÔT ATTRIBUÉE PAR L'INPI 15 NOV. 2002		1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE CABINET BALLOT 9, rue Claude Chappe Metz Technopôle 57070 METZ	
Vos références pour ce dossier (facultatif) 016721			
Confirmation d'un dépôt par télécopie		<input type="checkbox"/> N° attribué par l'INPI à la télécopie	
2 NATURE DE LA DEMANDE		Cochez l'une des 4 cases suivantes	
Demande de brevet		<input checked="" type="checkbox"/>	
Demande de certificat d'utilité		<input type="checkbox"/>	
Demande divisionnaire		<input type="checkbox"/>	
Demande de brevet initiale ou demande de certificat d'utilité initiale		N° _____ Date _____ N° _____ Date _____	
Transformation d'une demande de brevet européen Demande de brevet initiale		<input type="checkbox"/> N° _____ Date _____	
3 TITRE DE L'INVENTION (200 caractères ou espaces maximum) Procédé de division entière sécurisé contre les attaques à canaux cachés.			
4 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE		Pays ou organisation _____ N° _____ Date _____ Pays ou organisation _____ N° _____ Date _____ Pays ou organisation _____ N° _____ Date _____ <input type="checkbox"/> S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suite»	
5 DEMANDEUR (Cochez l'une des 2 cases)		<input checked="" type="checkbox"/> Personne morale <input type="checkbox"/> Personne physique	
Nom ou dénomination sociale		GEMPLUS	
Prénoms			
Forme juridique		Société Anonyme	
N° SIREN			
Code APE-NAF			
Domicile ou siège	Rue	Avenue du Pic de Bertagne Parc d'Activités de GEMENOS	
	Code postal et ville	[1][3][4][2][0] GEMENOS	
	Pays	FRANCE	
Nationalité		française	
N° de téléphone (facultatif)		N° de télécopie (facultatif)	
Adresse électronique (facultatif)			
<input type="checkbox"/> S'il y a plus d'un demandeur, cochez la case et utilisez l'imprimé «Suite»			

REMISE DES PIÈCES
DATE **15 NOV 2002**
LIEU **54 INPI NANCY**
N° D'ENREGISTREMENT **0214281**
NATIONAL ATTRIBUÉ PAR L'INPI

DB 540 W / 210502

6 MANDATAIRE (y compris le cas échéant)			
Nom	LECLAIRE		
Prénom	Jean-Louis		
Cabinet ou Société	CABINET BALLOT		
N° de pouvoir permanent et/ou de lien contractuel			
Adresse	Rue	9, rue Claude Chappe Metz Technopôle	
	Code postal et ville	57 10 17 10 METZ	
	Pays	FRANCE	
N° de téléphone (facultatif)	03.87.74.81.36		
N° de télécopie (facultatif)	03.87.36.26.76		
Adresse électronique (facultatif)			
7 INVENTEUR(S)		Les inventeurs sont nécessairement des personnes physiques	
Les demandeurs et les inventeurs sont les mêmes personnes		<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non : Dans ce cas remplir le formulaire de Désignation d'inventeur(s)	
8 RAPPORT DE RECHERCHE		Uniquement pour une demande de brevet (y compris division et transformation)	
Établissement immédiat ou établissement différé		<input checked="" type="checkbox"/> Établissement immédiat <input type="checkbox"/> Établissement différé	
Paiement échelonné de la redevance (en deux versements)		Uniquement pour les personnes physiques effectuant elles-mêmes leur propre dépôt <input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non	
9 RÉDUCTION DU TAUX DES REDEVANCES		Uniquement pour les personnes physiques <input type="checkbox"/> Requête pour la première fois pour cette invention (joindre un avis de non-imposition) <input type="checkbox"/> Obtenue antérieurement à ce dépôt pour cette invention (joindre une copie de la décision d'admission à l'assistance gratuite ou indiquer sa référence) : AG 	
10 SÉQUENCES DE NUCLEOTIDES ET/OU D'ACIDES AMINÉS		<input type="checkbox"/> Cochez la case si la description contient une liste de séquences	
Le support électronique de données est joint		<input type="checkbox"/> Oui <input type="checkbox"/> Non	
La déclaration de conformité de la liste de séquences sur support papier avec le support électronique de données est jointe		<input type="checkbox"/> Oui <input type="checkbox"/> Non	
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes			
11 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE (Nom et qualité du signataire) Jean-Louis LECLAIRE - 93.4009 		CABINET BALLOT CONSEILS EN PROPRIÉTÉ INDUSTRIELLE 9, rue Claude Chappe Technopôle Metz 2000 57070 METZ	
		VISA DE LA PRÉFECTURE OU DE L'INPI 	

PROCEDE DE DIVISION ENTIERE
SECURISE CONTRE LES ATTAQUES A CANAUX CACHES

L'invention concerne un procédé de division entière sécurisé contre les attaques de type à canal caché. L'invention est notamment intéressante pour réaliser des opérations de division dans un procédé cryptographique plus général, par exemple un procédé cryptographique à
5 clé secrète ou publique. Un tel procédé cryptographique peut par exemple être mis en œuvre dans des dispositifs électroniques tels que des cartes à puce.

10 La sécurité des procédés cryptographiques résident dans leur capacité à maintenir cachées les données confidentielles ou des données dérivées des données confidentielles qu'ils manipulent.

Un utilisateur malveillant peut éventuellement
15 engager des attaques, visant à découvrir notamment des données confidentielles contenues et manipulées dans des traitements effectués par le dispositif de calcul exécutant un procédé cryptographique.

Parmi les attaques les plus connues, on peut citer
20 les attaques à canaux cachés, simples ou différentielles. On entend par attaque à canal caché une attaque basée sur une grandeur physique mesurable de l'extérieur du dispositif, et dont l'analyse directe (attaque simple) ou l'analyse selon une méthode statistique (attaque
25 différentielle) permet de découvrir des données contenues et manipulées dans des traitements réalisés dans le dispositif. Ces attaques ont notamment été dévoilées par Paul Kocher (Advances in Cryptology - CRYPTO'99, vol. 1666 of Lecture Notes in Computer Science, pp.388-397.
30 Springer-Verlag, 1999).

Parmi les grandeurs physiques qui peuvent être exploitées à ces fins, on peut citer le temps

d'exécution, la consommation en courant, le champ électromagnétique rayonné par la partie du composant utilisée pour exécuter le calcul, etc. Ces attaques sont basées sur le fait que, au cours de l'exécution d'un
 5 procédé, la manipulation d'un bit, c'est à dire son traitement par une instruction particulière, laisse une empreinte particulière sur la grandeur physique considérée, selon la valeur de ce bit et / ou selon
 l'instruction.

10

Les procédés cryptographiques utilisant comme opération de base une opération d'exponentiation modulaire de type $Y = X^D$, X , Y et D étant des nombres entiers ont été très largement étudiés ces dernières
 15 années. A titre d'exemple, on peut citer le procédé RSA, l'échange de clé selon Diffie-Hellman ou le procédé de signature DSA. Des progrès significatifs ont été réalisés pour protéger ces procédés contre les attaques à canaux cachés.

20

Par contre, aucune étude n'a été faite sur la sécurisation des procédés cryptographiques utilisant comme opération élémentaire une division entière de type $q = a \text{ div } b$ et $r = a \text{ mod } b$, a et b étant deux opérandes, q et r étant respectivement le quotient et le reste de la
 25 division entière de a par b . a et / ou b sont des données secrètes, par exemple des éléments d'une clé du procédé. Par exemple, le procédé de Barrett (P. Barret, "Implementing the RSA public key encryption algorithm on a standard digital signal processing", vol 263 of Lecture
 30 Notes in Computer Science, pp. 311-323, Springer Verlag, 1987), le procédé de Quisquater (US patent 5,166,978, nov 92) ou le procédé RSA mis en œuvre selon le théorème des restes chinois (JJ Quisquater and C Couvreur, "Fast decipherment algorithm for RSA public key cryptosystem",
 35 Electronics Letters , vol 18, pp. 905-907, Octobre 1982)

sont des procédés cryptographiques utilisant une division entière comme opération élémentaire.

Un procédé connu pour mettre en œuvre une division entière est le procédé dit "papier crayon". Ce procédé reprend en pratique la méthode utilisée lorsqu'une telle opération est réalisée à la main. Ce procédé est rappelé ci-dessous.

Etant donné deux données $a = (a_{m-1}, \dots, a_0)$ de m bits et $b = (b_{n-1}, \dots, b_0)$ de n bits, n inférieur ou égal à m et $b_{n-1} \neq 0$, le procédé de division dit "papier crayon" calcule le quotient $q = a \text{ div } b$ et le reste $r = a \text{ div } b$. Pour cela, le procédé réalise successivement plusieurs divisions d'un entier A de $n+1$ bits par l'entier b de n bits. On doit avoir en pratique $0 \leq A/b < 2$, ce qui est le cas chaque fois que $b_{n-1} \neq 0$.

Le reste r est un nombre de au plus n bits puisque $r < b$. Le quotient q est quant à lui un nombre de au plus $m-n+1$ bits puisque $q = a \text{ div } b \leq a \text{ div } (b_{n-1} \cdot 2^{n-1}) = a \text{ div } 2^{n-1} = (a_{m-1}, \dots, a_{n-1})$ car $b \geq b_{n-1} \cdot 2^{n-1}$ et $(a_{m-1}, \dots, a_{n-1})$ est un nombre de $m-n+1$ bits. A la fin du procédé de division, le quotient q est mémorisé dans les $m-n+1$ bits de poids les plus faibles du registre contenant initialement le nombre a . Le bit de poids le plus fort du reste r est mémorisé dans un registre de 1 bit utilisé comme retenue (carry) pendant le calcul et les $n-1$ bits de poids les plus faibles du reste r sont mémorisés dans les $n-1$ bits de poids les plus forts du registre contenant initialement le nombre a .

Comme on travaille en base 2, le bit de quotient de la division entière $A \text{ div } b$ a seulement deux valeurs possibles : 0 ou 1. Aussi une manière simple de réaliser l'opération $A \text{ div } b$ consiste à soustraire b à A puis à tester le résultat : si le résultat de $A - b$ est positif, alors $A \text{ div } b = 1$, si le résultat de $A - b$ est strictement négatif, alors $A \text{ div } b = 0$.

Le procédé de division complet peut alors s'écrire de la manière suivante :

```

Entrée : a = (0, am-1, ..., a0)
5      b = (bn-1, ..., b0)
Sortie : q = a div b et r = a mod b
A = (0, am-1, ..., am-n+1)
Pour j = 1 à (m-n+1), faire :


---


10      a <- SHLm+1(a, 1) ; σ <- carry
      A <- SUBn(A, b) ; σ <- σ OU carry
      si (¬σ = VRAI) alors A <- ADDn(A, b)
      sinon lsb(a) = 1
Fin Pour

```

Procédé 1

15 Dans ce procédé, et dans tout ce qui suit, les notations suivantes sont utilisées.

Le symbole "<-" et la notation y <- x la notation est utilisé pour indiquer le chargement du contenu d'un registre contenant une donnée x dans un registre dont le contenu est appelé y.

20 A est un mot de n bits correspondant au contenu des n bits de poids les plus forts du registre contenant initialement la donnée a. A est bien sûr modifié à chaque itération.

σ indique si la soustraction a été effectuée à tort ou pas (ie si le bit de quotient doit être égal à 0 ou à 1).

30 ¬σ est le complément à 1 (encore appelé négation) de la variable σ. VRAI est une constante, égale à 1 dans un exemple.

lsb(a) est le bit de poids le plus faible du nombre a, également appelé bit le moins significatif de a.

35 SHL_{m+1}(a, 1) est une opération de décalage à gauche de 1 bit dans le registre de m+1 bits contenant la donnée a, le bit sortant du registre étant mémorisé dans la

variable carry et un bit égal à 0 étant entré en bit de poids le plus faible du registre contenant initialement la donnée a.

5 $ADD_n(A, b)$ est une opération d'addition des n bits du nombre b aux n bits du mot A. On notera que l'opération $SHL_n(a, 1)$ est équivalente à l'opération $ADD_n(a, a)$. Bien sûr l'addition $ADD_n(A, b)$ est réalisée en additionnant, dans un circuit d'addition de contenu de registre approprié, le contenu de deux registres
10 contenant respectivement A et b.

$SUB_n(A, b)$ est une opération de soustraction du nombre b au mot A. Bien sûr la soustraction $SUB_n(A, b)$ est réalisée en soustrayant, dans un circuit approprié, le contenu d'un registre contenant la donnée b au contenu du
15 registre contenant le mot A.

Enfin, par abus de langage mais surtout par souci de clarté, on utilisera le même nom pour parler d'un registre et de son contenu. Ainsi le registre A est en fait le registre contenant la donnée A.

20 En résumé, le procédé 1 réalise les étapes suivantes :

- si $a \leftarrow SHL_{m+1}(a, 1)$ génère une retenue ($\sigma = \text{carry} = 1$), cela signifie que $a_m = 1$ (avant décalage) et donc que b doit être soustrait à A.

25 - si $a_{m+1} = 0$ (avant décalage) et si $A \leftarrow SUB_n(A, b)$ génère une retenue ($\text{carry} = 1$), cela signifie que $A - b \geq 0$ avant la soustraction et donc b doit être soustrait à A.

- si $a \leftarrow SHL_{m+1}(a, 1)$ ne génère pas de retenue et
30 si $A \leftarrow SUB_n(A, b)$ ne génère pas non plus de retenue (c'est-à-dire si, après mise à jour de σ , σ est faux (ou $\neg\sigma$ est VRAI, alors cela signifie que $A - b < 0$ avant la soustraction et donc que b n'aurait pas dû être soustrait à A. Dans ce cas, le procédé réalise une opération
35 d'addition $A \leftarrow ADD_n(A, b)$ pour restaurer la valeur de A.

Le procédé 1 est sensible aux attaques à canal caché. En effet, on remarque sur le procédé 1 que, à chaque itération, selon la valeur de σ , c'est-à-dire selon la valeur du bit de quotient qui sera obtenu lors

5 de l'itération en cours, on effectue une addition $ADD_n(A, b)$ ou pas. Le nombre d'opérations effectuées au cours d'une itération varie donc en fonction du bit de résultat obtenu lors de ladite itération. Or, la

10 ou la durée de chaque itération varie en fonction du nombre d'opérations effectuées. En mesurant et en étudiant par exemple la trace laissée par le composant lors de l'exécution du procédé, il est alors possible de déterminer bit à bit la valeur des bits de résultat.

15

Un autre procédé également connu pour réaliser des division entière est une variante du procédé "papier-crayon", dite "sans restauration" (Non-Restoring Binary Division Algorithm, notamment décrit dans "J.J.F. Cavanagh, Digital Computer Arithmetic, Mac Graw-Hill

20 Company, 1984".

Entrée : $a = (0, a_{m-1}, \dots, a_0)$
 $b = (b_{n-1}, \dots, b_0)$
 Sortie : $q = a \text{ div } b$ et $r = a \text{ mod } b$

25 $\sigma' \leftarrow 1$; $A = (0, a_{m-1}, \dots, a_{m-n+1})$
 Pour $j = 1$ à $(m-n+1)$, faire :

$a \leftarrow SHL_{m+1}(a, 1)$; $\sigma \leftarrow \text{carry}$
 si $(\sigma' = \text{VRAI})$ alors $A \leftarrow SUB_n(A, b)$
 $\sigma \leftarrow \sigma \text{ OU } \text{carry}$
 30 sinon $A \leftarrow ADD_n(A, b)$
 $\sigma \leftarrow \sigma \text{ ET } \text{carry}$
 si $(\sigma = \text{VRAI})$ alors $lsb(a) = 1$

 $\sigma' \leftarrow \sigma$

Fin Pour

35 si $(\neg \sigma = \text{VRAI})$ alors $A \leftarrow ADD_n(A, b)$

Procédé 2

Par rapport au procédé 1, le procédé utilise une nouvelle variable σ' pour conserver la valeur de σ obtenue à l'itération précédente. Ici, selon la valeur de σ , on effectue une addition ou une soustraction. Dit autrement, si au cours d'une itération, b est soustrait à tort à A , alors la valeur de A est restaurée au cours de l'itération suivante, et non plus à la fin de l'itération en cours comme dans le cas du procédé 1.

Quelle que soit la valeur de σ au cours d'une itération, le procédé réalise le même nombre d'opérations au cours de chaque itération. Cette précaution n'est cependant pas suffisante pour protéger le procédé contre les attaques à canal caché. En effet, à chaque itération, on réalise une opération de décalage $a \leftarrow \text{SHL}_{m+1}(a, 1)$ puis, selon la valeur de σ , une addition $A \leftarrow \text{ADD}_n(A, b)$ ou une soustraction $A \leftarrow \text{SUB}_n(A, b)$.

Or, la réalisation d'une soustraction est plus longue et consomme plus d'énergie que la réalisation d'une opération d'addition. En effet, le plus souvent, les moyens de calcul utilisés pour mettre en œuvre le procédé ne comprennent pas de circuit de soustraction. L'opération de soustraction est réalisée en calculant d'abord le complément à 2^n de b , noté \bar{b} , puis en additionnant \bar{b} à A , la retenue éventuelle de l'addition étant mémorisée dans la variable carry. Ce mode de réalisation d'une soustraction est justifié par le fait que, par définition de \bar{b} , on a $b + \bar{b} = 2^n$. On a donc $A - b = A + \bar{b} - 2^n = A + \bar{b} \bmod (2^n)$, $\bmod (2^n)$ étant une réduction modulo 2^n . Deux opérations, une opération de complément à 2^n et une addition, sont donc en pratique nécessaires pour réaliser une soustraction.

Comme les procédés connus de division entière ne sont pas protégés contre les attaques à canal caché, tout procédé cryptographique utilisant les procédés de

division entière connu ne sont donc pas plus protégés contre de telles attaques à canal caché.

Par ailleurs, statistiquement, 50% des bits du quotient obtenu par un procédé de division sont égaux à 0, ce qui signifie que statistiquement, le procédé
 5 0, ce qui signifie que statistiquement, le procédé compense une soustraction sur deux faite à tort. Le temps d'exécution du procédé 1 est donc statistiquement 1,5 fois plus long que le temps d'exécution du procédé 2.

10 Au vu des problèmes des procédés cryptographiques actuels, un objet essentiel de l'invention est un nouveau procédé de réalisation d'une division entière, protégé contre les attaques à canal caché.

Un objet supplémentaire de l'invention est un
 15 procédé de réalisation d'une division entière dont le temps d'exécution est très faible.

Un objet supplémentaire également de l'invention est un procédé de réalisation d'une division entière au cours duquel seul le registre contenant la donnée
 20 initiale a est modifié, remplacé par le quotient et le résultat, tout autre registre de la mémoire (et notamment le registre contenant initialement la donnée b) restant inchangé à la fin de l'exécution du procédé.

Avec cet objectif principal et ces objectifs
 25 subsidiaires en vue, l'invention propose un procédé cryptographique au cours duquel on réalise une division entière de type $q = a \text{ div } b$ et $r = a \text{ mod } b$, avec a un nombre de m bits, b un nombre de n bits avec n inférieur ou égal à m et b_{n-1} non nul, b_{n-1} étant le bit de poids le
 30 plus fort de b, procédé au cours duquel, à chaque itération d'une boucle indicée par i variant entre 1 et m-n+1, on réalise une division partielle d'un mot A de n bits du nombre a par le nombre b pour obtenir un bit du quotient q.

Selon l'invention, les mêmes opérations sont réalisées à chaque itération, quelque soit la valeur du bit de quotient obtenu.

5 Ainsi, avec le procédé selon l'invention, il n'est plus possible de déterminer les bits du résultat à partir de la trace laissée lors de l'exécution du procédé de l'invention.

10 Selon un premier mode de réalisation du procédé de l'invention, à chaque itération, on réalise une opération d'addition du nombre b au mot A et une soustraction du nombre b au mot A .

Selon ce premier mode de réalisation, le procédé comprend de préférence l'ensemble des étapes suivantes :

15 Entrée : $a = (0, a_{m-1}, \dots, a_0)$
 $b = (b_{n-1}, \dots, b_0)$
 Sortie : $q = a \text{ div } b$ et $r = a \text{ mod } b$
 $\sigma' \leftarrow 1$; $A = (0, a_{m-1}, \dots, a_{m-n+1})$
 Pour $j = 1$ à $(m-n+1)$, faire :
 20 $a \leftarrow \text{SHL}_{m+1}(a, 1)$; $\sigma \leftarrow \text{carry}$
 $A \leftarrow (\sigma') \text{SUB}_n(A, b) + (\neg \sigma') \text{ADD}_n(A, b)$
 $\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET } \text{carry}) \oplus (\sigma' \text{ ET } \text{carry})$
 $\text{lsb}(a) \leftarrow \sigma$
 $\sigma' \leftarrow \sigma$
 Fin Pour
 25 si $(\neg \sigma = \text{VRAI})$ alors $A \leftarrow \text{ADD}_n(A, b)$

Dans ce mode de réalisation, la variable carry ci-dessus désigne la retenue résultant de l'opération $\text{SUB}_n(A, b)$ lorsque σ' vaut 1 et la retenue résultant de l'opération $\text{ADD}_n(A, b)$ lorsque σ' vaut 0.

30

Selon un deuxième mode de réalisation du procédé selon l'invention, à chaque itération, on réalise une opération d'addition soit du nombre b soit d'un nombre \bar{b} complémentaire du nombre b avec le mot A .

35 De préférence, au cours de chaque itération, on réalise également une mise à jour d'une première variable

(σ') en fonction du bit du quotient produit, la dite première variable (σ') indiquant si, lors de l'itération suivante, le nombre b ou le nombre \bar{b} doit être additionné au mot A .

- 5 De préférence encore, selon ce mode de réalisation, le procédé comprend l'ensemble des étapes suivantes :

Entrée : $a = (0, a_{m-1}, \dots, a_0)$

$b = (b_{n-1}, \dots, b_0)$

~~Sortie : $q = a \text{ div } b$ et $r = a \text{ mod } b$~~

10 $A = (0, a_{m-1}, \dots, a_{m-n+1})$; $\sigma' \leftarrow 1$; $\bar{b} \leftarrow \text{CPL}_{2n}(b)$

Pour $j = 1$ à $(m-n+1)$, faire :

$a \leftarrow \text{SHL}_{m+1}(a, 1)$; $\sigma \leftarrow \text{carry}$

$d_{\text{addr}} \leftarrow b_{\text{addr}} + \sigma' (\bar{b}_{\text{addr}} - b_{\text{addr}})$

$A \leftarrow \text{ADD}_n(A, d)$

15 $\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET } \text{carry}) \oplus (\sigma' \text{ ET } \text{carry})$

$\text{lsb}(a) \leftarrow \sigma$

$\sigma' \leftarrow \sigma$

Fin Pour

si $(\neg \sigma = \text{VRAI})$ alors $A \leftarrow \text{ADD}_n(A, b)$

20

Selon un troisième mode de réalisation du procédé selon l'invention, à chaque itération, on réalise une opération de complément à 2^n d'une donnée actualisée (b ou \bar{b}) ou d'une donnée fictive (c ou \bar{c}) puis une opération d'addition de la donnée actualisée avec le mot A .

25

De préférence, au cours de chaque itération, on réalise également à chaque itération une mise à jour d'une deuxième variable (δ) en fonction du bit du quotient produit, la dite deuxième variable (δ) indiquant si, lors de l'itération suivante, l'opération de complément à $2n$ doit être réalisée sur la donnée actualisée ou sur la donnée fictive.

30

De préférence encore, au cours de chaque itération, on réalise également à chaque itération la mise à jour d'une troisième variable (β) indiquant si la donnée

35

actualisée est égale au nombre b ou au nombre complémentaire \bar{b} .

De préférence encore, selon ce mode de réalisation, le procédé comprend l'ensemble des étapes suivantes :

```

5      Entrée :  $a = (0, a_{m-1}, \dots, a_0)$ 
            $b = (b_{n-1}, \dots, b_0)$ 
      Sortie :  $q = a \text{ div } b$  and  $r = a \text{ mod } b$ 
       $\sigma' \leftarrow 1$  ;  $\beta \leftarrow 1$ ,  $\gamma \leftarrow 1$  ;  $A = (0, a_{m-1}, \dots, a_{m-n+1})$ 
      pour  $j = 1$  à  $(m-n+1)$  faire
10       $a \leftarrow \text{SHL}_{m+1}(a, 1)$  ;  $\sigma \leftarrow \text{carry}$ 
            $\delta \leftarrow \sigma' \oplus \beta$ 
            $d_{\text{addr}} \leftarrow b_{\text{addr}} + \delta(C_{\text{addr}} - b_{\text{addr}})$ 
            $d \leftarrow \text{CPL}_{2n}(d)$ 
            $A \leftarrow \text{ADD}_n(A, b)$ 
15       $\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET } \text{carry}) \oplus (\sigma' \text{ ET } \text{carry})$ 
            $\beta \leftarrow \neg \sigma'$  ;  $\gamma \leftarrow \gamma \oplus \delta$  ;  $\sigma' \leftarrow \sigma$ 
            $\text{lsb}(a) = \sigma$ 
      fin pour
      si  $(\neg \beta = \text{VRAI})$  alors  $b \leftarrow \text{CPL}_{2n}(b)$ 
20      si  $(\neg \gamma = \text{VRAI})$  alors  $c \leftarrow \text{CPL}_{2n}(c)$ 
      si  $(\neg \sigma = \text{VRAI})$  alors  $A \leftarrow \text{ADD}_n(A, b)$ 

```

L'invention concerne également un composant électronique comprenant des moyens de calcul programmés pour mettre en œuvre un procédé tel que décrit ci-dessus, les moyens de calcul comprenant notamment une unité centrale associée à une mémoire comprenant plusieurs registres pour mémoriser les données a et b .

Enfin, l'invention concerne également une carte à puce comprenant un circuit intégré tel que décrit ci-dessus.

L'invention sera mieux comprise et d'autres caractéristiques et avantages apparaîtront à la lecture de la description qui va suivre, d'exemples de

réalisation de procédés de division entière selon l'invention.

Dans un 1^{er} exemple de mise en œuvre de l'invention,
 5 on réalise un procédé sécurisé contre les attaques à canal caché en supprimant les opérations de test (de type si ... alors ... si non ...) du procédé 2 et donc les conséquences de leur présence.

~~Selon l'invention, on remplace, dans le procédé 2,~~
 10 les étapes si ... alors ... sinon par les trois étapes suivantes :

$A \leftarrow \sigma' \text{SUB}_n(A, b) + (\neg \sigma') \text{ADD}_n(A, b)$
 $\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET carry}) \oplus (\sigma' \text{ ET carry})$
 $\text{lsb}(a) \leftarrow \sigma$

15 On obtient ainsi le procédé selon l'invention suivant :

Entrée : $a = (0, a_{m-1}, \dots, a_0)$
 $b = (b_{n-1}, \dots, b_0)$

20 Sortie : $q = a \text{ div } b$ et $r = a \text{ mod } b$

$A = (0, a_{m-1}, \dots, a_{m-n+1})$; $\sigma' \leftarrow 1$

Pour $j = 1$ à $(m-n+1)$, faire :

$a \leftarrow \text{SHL}_{m+1}(a, 1)$; $\sigma \leftarrow \text{carry}$
 $A \leftarrow (\sigma') \text{SUB}_n(A, b) + (\neg \sigma') \text{ADD}_n(A, b)$
 25 $\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET carry}) \oplus (\sigma' \text{ ET carry})$
 $\text{lsb}(a) \leftarrow \sigma$
 $\sigma' \leftarrow \sigma$

Fin Pour

si $(\neg \sigma = \text{VRAI})$ alors $A \leftarrow \text{ADD}_n(A, b)$

30 Procédé 3

Le procédé 3 est équivalent au procédé 2 en ce sens qu'il produit le même résultat à partir des mêmes données a et b d'entrée. En effet, dans le procédé 2, lorsque
 35 $\sigma' = 1$, on réalise l'opération $A \leftarrow \text{SUB}_n(A, b)$ et lorsque $\sigma' = 0$, on réalise l'opération $A \leftarrow \text{ADD}_n(A, b)$. Il en est

de même dans le procédé 3 puisque $\sigma' = \neg(\neg\sigma')$. Par ailleurs, dans le procédé 2, lorsque $\sigma' = 1$ on réalise l'opération $\sigma \leftarrow \sigma \text{ OU } \text{carry}$, et lorsque $\sigma' = 0$ on réalise l'opération $\sigma \leftarrow \sigma \text{ ET } \text{carry}$. Ceci peut s'écrire sous la

5 forme

$\sigma \leftarrow (\sigma')(\sigma \text{ OU } \text{carry}) + (\neg\sigma')(\sigma \text{ ET } \text{carry}),$
ce qui est logiquement équivalent à

$\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET } \text{carry}) \oplus (\sigma' \text{ ET } \text{carry})$

Enfin, dans le procédé 2, en réalisant l'opération

10 $a \leftarrow \text{SHL}_{m+1}(a, 1)$, on fixe à 0 le bit de poids le plus faible de a (dit autrement $\text{lsb}(a) = 0$) puis, à la fin de l'itération en cours, si $\sigma = 1$, on réalise l'opération $\text{lsb}(a) = 1$, sinon, si $\sigma = 0$, $\text{lsb}(a)$ n'est pas modifié. On peut donc aisément remplacer l'opération {si $\sigma = 1$,

15 $\text{lsb}(a) = 1$ } par l'opération $\text{lsb}(a) = \sigma$, quelle que soit la valeur de σ .

Le procédé 3 est non seulement équivalent au procédé 2 mais il est également sûr vis-à-vis des attaques à canal caché. En effet, le procédé ne contient

20 aucune opération de test de type si ... alors ... sinon, et les mêmes opérations sont réalisées à chaque itération, quels que soient le bit de la donnée d'entrée utilisé et / ou le bit de résultat obtenu au cours d'une itération. Il est donc impossible, à partir de la trace laissée par

25 le composant, de séparer les différentes itérations et de déterminer les bits de la donnée d'entrée et / ou de la donnée de sortie.

Dans un 2^{ème} exemple de mise en œuvre de

30 l'invention, on modifie le procédé 3 selon l'invention en limitant de plus le temps d'exécution du procédé.

Comme on l'a vu précédemment, pour réaliser une opération de soustraction $A \leftarrow \text{SUB}_n(A, b)$, on réalise en pratique une opération $\bar{b} = \text{CPL}_{2n}(b)$ de complément à 2^n du

35 nombre b puis une opération d'addition de type $A \leftarrow \text{ADD}_n(A, \bar{b})$.

Ce qui signifie, pour le procédé 3, qu'à chaque itération une opération de complément à 2^n est réalisée, en plus d'une opération d'addition $A \leftarrow \text{ADD}_n(A, b)$ ou $A \leftarrow \text{ADD}_n(A, \bar{b})$.

- 5 Pour diminuer le temps d'exécution, on limite le nombre d'opérations de complément à 2^n $\bar{b} \leftarrow \text{CPL}_{2^n}(b)$, on utilise un espace mémoire additionnel pour stocker au début du procédé la valeur de \bar{b} . Il suffit alors
-
- 10 d'ajouter b à A pour effectuer $A \leftarrow \text{ADD}_n(A, b)$. Cela permet également de réaliser une seule opération d'addition par itération, de sorte que la vitesse d'exécution est encore augmentée.

- On utilise ici deux registres b et \bar{b} pour mémoriser
- 15 respectivement les données b et \bar{b} et ayant pour adresse b_{addr} et \bar{b}_{addr} . On appelle d le registre dont le contenu est additionné au contenu du registre A au cours d'une itération donnée et on appelle d_{addr} son adresse. En
- 20 pratique, à chaque itération, le registre d est soit le registre contenant b soit le registre contenant \bar{b} . Comme dans le procédé 3, la variable σ' est utilisée pour garder une trace de ce qui s'est passé au cours d'une itération donnée et déterminer si une addition ou une
- 25 soustraction doit être réalisée à l'itération suivante. En regroupant le tout, on obtient finalement le procédé 4 suivant :

- Entrée : $a = (0, a_{m-1}, \dots, a_0)$
 $b = (b_{n-1}, \dots, b_0)$
- 30 Sortie : $q = a \text{ div } b$ et $r = a \text{ mod } b$
 $A = (0, a_{m-1}, \dots, a_{m-n+1})$; $\sigma' \leftarrow 1$; $\bar{b} \leftarrow \text{CPL}_{2^n}(b)$
 Pour $j = 1$ à $(m-n+1)$, faire :
-
- 35 $a \leftarrow \text{SHL}_{m+1}(a, 1)$; $\sigma \leftarrow \text{carry}$
 $d_{\text{addr}} \leftarrow b_{\text{addr}} + \sigma' (\bar{b}_{\text{addr}} - b_{\text{addr}})$
 $A \leftarrow \text{ADD}_n(A, d)$
 $\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET } \text{carry}) \oplus (\sigma' \text{ ET } \text{carry})$

lsb(a) <- σ

$\sigma' <- \sigma$

Fin Pour

si ($\neg\sigma = \text{VRAI}$) alors A <- $\text{ADD}_n(A, b)$

5

Procédé 4

Dans un 3^{ème} exemple de mise en œuvre de l'invention, on modifie le procédé 4 selon l'invention en limitant l'espace mémoire utilisé pour mettre en œuvre le procédé.

10

Pour cela, la valeur \bar{b} complémentaire de b résultat de l'opération $\text{CPL}_{2n}(b)$ est mémorisée à la place de la valeur initiale de b, dans le même registre. L'opération de soustraction est ainsi réalisée en remplaçant b par son complément \bar{b} dans le même registre puis en additionnant à A le contenu du dit registre.

15

De plus, on évite le calcul de valeurs inutiles de \bar{b} (c'est le cas lorsque deux itérations successives j et j+1 utilisent toutes deux la même addition soit A <- A+b soit A <- A + \bar{b}). Pour cela, on utilise un autre registre c dont le contenu, indifférent ou fictif, est remplacé par son complément à 2^n lorsqu'il n'est pas nécessaire de remplacer le contenu du registre contenant initialement b (c'est-à-dire lorsque deux itérations successives utilisent soit b soit \bar{b}). En pratique, le registre c est un registre quelconque de la mémoire, de même taille que le registre contenant b, mais différent des registres contenant initialement a ou b. Le registre c peut être utilisé par ailleurs pour réaliser d'autres opérations. A la fin du procédé de l'invention, le registre c contient sa valeur initiale, c'est-à-dire celle qu'il avait avant exécution du procédé. La valeur initiale du contenu du registre c est totalement indifférente car cette valeur n'est pas réellement utilisée dans le cadre du procédé

20

25

30

35

selon l'invention.

On appelle d_{addr} l'adresse du registre contenant la valeur qui sera remplacée par son complément à 2^n lors de l'itération en cours : d_{addr} est soit b_{addr} si le contenu du registre contenant initialement b doit être complémenté à 2^n , soit c_{addr} sinon. On appelle d le contenu du registre dont l'adresse est d_{addr} .

On utilise également des variables β et γ pour garder une trace de l'état de la valeur contenue dans les registres localisés à l'adresse b_{addr} et c_{addr} . Cet état est soit la valeur originale soit la valeur originale complémentée à 2^n . On choisit $\beta = 1$ (resp. $\gamma = 1$) lorsque la valeur localisée à l'adresse b_{addr} (resp. c_{addr}) est la valeur originale, et $\beta = 0$ (resp. $\gamma = 0$) lorsque la valeur localisée à l'adresse b_{addr} (resp. c_{addr}) est le complément à 2^n de la valeur originale. La variable σ' est utilisée pour garder une trace de la valeur de la variable σ à l'itération précédente. Comme précédemment, $\sigma' = 0$ signifie qu'une soustraction ($A \leftarrow SUB_n(A, b) = ADD_n(A, \bar{b})$) non nécessaire a été effectuée à l'itération précédente et qu'une opération d'addition $A \leftarrow ADD_n(A, b)$ doit être réalisée pendant l'itération en cours pour compenser. Inversement, $\sigma' = 1$ signifie qu'aucune soustraction n'a été effectuée à tort lors de l'itération précédente et qu'une soustraction doit être effectuée lors de l'itération en cours.

On obtient la table de vérité suivante :

valeurs précédentes			valeurs actualisées		
σ'	β	γ	β	γ	
0	0	0	1	0	
0	0	1	1	1	
0	1	0	1	1	
0	1	1	1	0	
1	0	0	0	1	
1	0	1	0	0	
1	1	0	0	0	
1	1	1	0	1	

On en déduit :

$$\beta \leftarrow \neg \sigma'$$

$$\gamma \leftarrow \gamma \oplus \sigma' \oplus \beta$$

5 En regroupant le tout on obtient finalement le procédé 5 suivant :

Entrée : $a = (0, a_{m-1}, \dots, a_0)$

$b = (b_{n-1}, \dots, b_0)$

10 Sortie : $q = a \text{ div } b$ and $r = a \text{ mod } b$

$\sigma' \leftarrow 1$; $\beta \leftarrow 1$, $\gamma \leftarrow 1$; $A = (0, a_{m-1}, \dots, a_{m-n+1})$

pour $j = 1$ à $(m-n+1)$ faire

$a \leftarrow \text{SHL}_{m+1}(a, 1)$; $\sigma \leftarrow \text{carry}$

$\delta \leftarrow \sigma' \oplus \beta$

15 $d_{\text{addr}} \leftarrow b_{\text{addr}} + \delta(C_{\text{addr}} - b_{\text{addr}})$

$d \leftarrow \text{CPL}_{2n}(d)$

$A \leftarrow \text{ADD}_n(A, b)$

$\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET } \text{carry}) \oplus (\sigma' \text{ ET } \text{carry})$

$\beta \leftarrow \neg \sigma'$; $\gamma \leftarrow \gamma \oplus \delta$; $\sigma' \leftarrow \sigma$

20 $\text{lsb}(a) = \sigma$

fin pour

si $(\neg \beta = \text{VRAI})$ alors $b \leftarrow \text{CPL}_{2n}(b)$

si $(\neg \gamma = \text{VRAI})$ alors $c \leftarrow \text{CPL}_{2n}(c)$

si $(\neg \sigma = \text{VRAI})$ alors $A \leftarrow \text{ADD}_n(A, b)$

25 Procédé 5

De manière générale, l'avantage essentiel de l'invention par rapport aux autres procédés connus réalisant la même opération est qu'il est sûr vis à vis
30 des attaques à canal caché, et notamment des attaques de type SPA. De plus, pour être mis en œuvre, le procédé selon l'invention ne demande pas plus de ressources (notamment en terme de temps d'exécution et d'espace mémoire) que les procédés connus de division entière, non
35 protégés.

REVENDEICATIONS

1. Procédé cryptographique au cours duquel on réalise une division entière de type $q = a \text{ div } b$ et $r = a \text{ mod } b$, avec q un quotient, a un nombre de m bits, b un nombre de n bits avec n inférieur ou égal à m et b_{n-1} non nul, b_{n-1} étant le bit de poids le plus fort de b , procédé
 5 au cours duquel, à chaque itération d'une boucle indicée

par i variant entre 1 et $m-n+1$, on réalise une division partielle d'un mot A de n bits du nombre a par le nombre b pour obtenir un bit du quotient q ,

10 le procédé étant caractérisé en ce que les mêmes opérations sont réalisées à chaque itération, quelque soit la valeur du bit de quotient obtenu.

2. Procédé selon la revendication 1, au cours
 15 duquel, à chaque itération, on réalise une addition du nombre b au mot A et une soustraction du nombre b au mot A .

3. Procédé selon l'une des revendications 1 à 2, au
 20 cours duquel on réalise l'ensemble des étapes suivantes :

Entrée : $a = (0, a_{m-1}, \dots, a_0)$

$b = (b_{n-1}, \dots, b_0)$

Sortie : $q = a \text{ div } b$ et $r = a \text{ mod } b$

$A = (0, a_{m-1}, \dots, a_{m-n+1})$; $\sigma' \leftarrow 1$

25 Pour $j = 1$ à $(m-n+1)$, faire :

$a \leftarrow \text{SHL}_{m+1}(a, 1)$; $\sigma \leftarrow \text{carry}$

$A \leftarrow (\sigma') \text{SUB}_n(A, b) + (\neg \sigma') \text{ADD}_n(A, b)$

$\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET } \text{carry}) \oplus (\sigma' \text{ ET } \text{carry})$

$\text{lsb}(a) \leftarrow \sigma$

30 $\sigma' \leftarrow \sigma$

Fin Pour

si $(\neg \sigma = \text{VRAI})$ alors $A \leftarrow \text{ADD}_n(A, b)$

4. Procédé selon la revendication 1, au cours duquel, à chaque itération, on réalise une opération d'addition soit du nombre b ou soit d'un nombre \bar{b} complémentaire du nombre b avec le mot A .

5

5. Procédé selon la revendication 4, au cours duquel, à chaque itération, on réalise également une mise à jour d'une première variable (σ') indiquant si, lors de l'itération suivante, le nombre b ou le nombre \bar{b} doit être additionné avec le mot A selon le bit de quotient produit ($\text{lsb}(a)$).

6. Procédé selon la revendication 4 ou la revendication 5, au cours duquel on réalise l'ensemble des étapes suivantes :

Entrée : $a = (0, a_{m-1}, \dots, a_0)$
 $b = (b_{n-1}, \dots, b_0)$
 Sortie : $q = a \text{ div } b$ et $r = a \text{ mod } b$
 $A = (0, a_{m-1}, \dots, a_{m-n+1})$; $\sigma' \leftarrow 1$; $\bar{b} \leftarrow \text{CPL}2_n(b)$
 Pour $j = 1$ à $(m-n+1)$, faire :
 $a \leftarrow \text{SHL}_{m+1}(a, 1)$; $\sigma \leftarrow \text{carry}$
 $d_{\text{addr}} \leftarrow b_{\text{addr}} + \sigma'(\bar{b}_{\text{addr}} - b_{\text{addr}})$
 $A \leftarrow \text{ADD}_n(A, d)$
 $\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET } \text{carry}) \oplus (\sigma' \text{ ET } \text{carry})$
 $\text{lsb}(a) \leftarrow \sigma$
 $\sigma' \leftarrow \sigma$
 Fin Pour
 si $(\neg \sigma = \text{VRAI})$ alors $A \leftarrow \text{ADD}_n(A, b)$

7. Procédé selon la revendication 1, au cours duquel, à chaque itération, on réalise une opération de complément à 2^n d'une donnée actualisée (b ou \bar{b}) ou d'une donnée fictive (c ou \bar{c}) puis une opération d'addition de la donnée actualisée avec le mot A .

35

8. Procédé selon la revendication 7, au cours duquel on réalise également à chaque itération une opération de mise à jour d'une deuxième variable (δ) indiquant si, lors de l'itération suivante, l'opération de complément à 2^n doit être réalisée sur la donnée actualisée ou sur la donnée fictive.

9. Procédé selon l'une des revendications 7 ou 8, ~~dans lequel on réalise également à chaque itération, une~~
mise à jour d'une troisième variable (β) indiquant si la donnée actualisée est égale à la donnée b ou à son complément à 2^n .

10. Procédé selon l'une des revendications 7 à 9, au cours duquel on réalise l'ensemble des étapes suivantes :

Entrée : $a = (0, a_{m-1}, \dots, a_0)$
 $b = (b_{n-1}, \dots, b_0)$
 Sortie : $q = a \text{ div } b$ and $r = a \text{ mod } b$
 $\sigma' \leftarrow 1$; $\beta \leftarrow 1$, $\gamma \leftarrow 1$; $A = (0, a_{m-1}, \dots, a_{m-n+1})$
 pour $j = 1$ à $(m-n+1)$ faire
 $a \leftarrow \text{SHL}_{m+1}(a, 1)$; $\sigma \leftarrow \text{carry}$
 $\delta \leftarrow \sigma' \oplus \beta$
 $d_{\text{addr}} \leftarrow b_{\text{addr}} + \delta(C_{\text{addr}} - b_{\text{addr}})$
 $d \leftarrow \text{CPL}_{2n}(d)$
 $A \leftarrow \text{ADD}_n(A, b)$
 $\sigma \leftarrow (\sigma \text{ ET } \sigma') \oplus (\sigma \text{ ET } \text{carry}) \oplus (\sigma' \text{ ET } \text{carry})$
 $\beta \leftarrow \neg \sigma'$; $\gamma \leftarrow \gamma \oplus \delta$; $\sigma' \leftarrow \sigma$
 $\text{lsb}(a) = \sigma$
 fin pour
 si $(\neg \sigma = \text{VRAI})$ alors $A \leftarrow \text{ADD}_n(A, b)$

11. Procédé selon la revendication 10, au cours duquel on réalise, à la fin, les opérations suivantes :
 si $(\neg \beta = \text{VRAI})$ alors $b \leftarrow \text{CPL}_{2n}(b)$
 si $(\neg \gamma = \text{VRAI})$ alors $c \leftarrow \text{CPL}_{2n}(c)$

12. Composant électronique comprenant des moyens de calcul programmés pour mettre en œuvre un procédé selon l'une des revendications 1 à 11, les moyens de calcul comprenant notamment une unité centrale associée à une
5 mémoire comprenant plusieurs registres pour mémoriser les données a et b.

13. Carte à puce comprenant un circuit intégré selon la revendication 12.

DÉPARTEMENT DES BREVETS

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08

Téléphone : 33 (1) 53 04 53 04 Télécopie : 33 (1) 42 94 86 54

DÉSIGNATION D'INVENTEUR(S) Page N° 1../1..

(À fournir dans le cas où les demandeurs et les inventeurs ne sont pas les mêmes personnes)



Cet imprimé est à remplir lisiblement à l'encre noire

DB 113 @ W / 270601

Vos références pour ce dossier (facultatif)		016721
N° D'ENREGISTREMENT NATIONAL		0214 281
TITRE DE L'INVENTION (200 caractères ou espaces maximum)		
Procédé de division entière sécurisé contre les attaques à canaux cachés.		
LE(S) DEMANDEUR(S) :		
GEMPLUS Avenue du Pic de Bertagne Parc d'activités de Gemenos 13420 GEMENOS FRANCE		
DESIGNE(NT) EN TANT QU'INVENTEUR(S) :		
<input checked="" type="checkbox"/> 1	Nom	JOYE
	Prénoms	Marc
Adresse	Rue	19, rue Voltaire
	Code postal et ville	18 316 410 SAINT-ZACHARIE
Société d'appartenance (facultatif)		
<input checked="" type="checkbox"/> 2	Nom	VILLEGAS
	Prénoms	Karine
Adresse	Rue	162, Chemin de Lieutaud
	Code postal et ville	13 412 10 GEMENOS
Société d'appartenance (facultatif)		
<input type="checkbox"/> 3	Nom	
	Prénoms	
Adresse	Rue	
	Code postal et ville	
Société d'appartenance (facultatif)		
S'il y a plus de trois inventeurs, utilisez plusieurs formulaires. Indiquez en haut à droite le N° de la page suivi du nombre de pages.		
DATE ET SIGNATURE(S) DU (DES) DEMANDEUR(S) OU DU MANDATAIRE (Nom et qualité du signataire) Jean-Louis LECLAIRE - 93.4009 		
CABINET BALLOT CONSEILS EN PROPRIÉTÉ INDUSTRIELLE 9, rue Claude Chappe Technopôle Metz 2000 57070 METZ		

PCT Application
PCT/FR2003/050119



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.